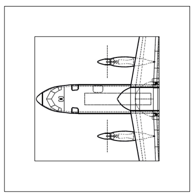
Institute of Scientific Computing
Carl-Friedrich-Gauss-Fakultät

Braunschweig, Germany

Martin Krosche   Wolfgang Heinze

A Robustness Analysis of a Preliminary Design of a CESTOL Aircraft



Informatikbericht 2014–02

This document was created in February 2014 using $\LaTeX 2_\varepsilon$.

Institute of Scientific Computing
Technische Universität Braunschweig
Hans-Sommer-Straße 65
D-38106 Braunschweig, Germany

e-mail: wire@tu-bs.de
url: https://www.tu-braunschweig.de/wire/

A Robustness Analysis of a Preliminary Design of a CESTOL Aircraft

Informatikbericht 2014–02

Martin Krosche Wolfgang Heinze*

Institute of Scientific Computing, TU Braunschweig

Abstract

As part of the Collaborative Research Center 880 preliminary aircraft design activities are carried out for a new class of low-noise cruise-efficient transport aircrafts with short take-off and landing capabilities (CESTOL). A corresponding aircraft is quite different from a state-of-the-art commercial aircraft because of the use of active high-lift devices. The fact that new technologies are not sufficiently understood yet in combination with the assumption of common design data and the use of classical calculation methods expresses itself in uncertainties which are of epistemic character. The robustness of a deterministic CESTOL aircraft design towards parameters such as the necessary engine thrust, direct operating costs, or the runway lengths is investigated here concerning the mentioned uncertainties. For this purpose a stochastic description of parameter variations of the design is formulated. Stochastic quantities are computed by Monte Carlo (MC) sampling to rate the robustness. A distributed component-based software implementation is used to perform the MC sampling. The software system is installed on a Linux cluster with several multi-CPU computers; a deterministic sample is simulated through the design program PrADO.

Keywords: aircraft design, CESTOL, internally blown flaps (Coanda), robustness analysis, Monte Carlo method, distributed software components, SFB 880

*Institute of Aircraft Design and Lightweight Structures (IFL), TU Braunschweig.

Contents

1 Introduction

Uncertainty enters the process of aircraft design in many places [1–4]. As a consequence, the finally manufactured aircraft differs from its aircraft design generated in the preliminary design phase. A prediction of the influence of these differences can be made by a systematic variation of design parameters. The variations can be formulated by a set of random variables which transfer a deterministic reference model of an aircraft design to a probabilistic model. Statistics on the probabilistic model are often obtained by sampling techniques like the basic Monte Carlo (MC) method [2, 5–9], importance sampling [10, 11] or latin hypercube sampling [12] to get an understanding about the robustness of the reference model. However, the simulation of a single sample is usually computationally expensive, and many of these simulations are required for accurate statistics. Alternatively, a surrogate model may be determined [7–11, 13] from which the statistics are obtained in a post-processing step. The surrogate models in these publications are described by a regression equation which typically consists of linear and quadratic terms and second order interaction terms of the random variables. A more recent representation of a surrogate model, which is not so common yet in preliminary aircraft design, is through a polynomial chaos expansion (PCE) [14–16], at which the probabilistic model of the aircraft design is projected onto a set of polynomials defined on independent random variables. The PCE may be called sparse [17–21] when only few polynomials are used for an accurate description of the model. Furthermore, a low-rank representation of the PC coefficients may be aimed to gain computational advantages [21–24]. A surrogate model approach may be more efficient than the other mentioned techniques. Overviews about numerical methods for the quantification of uncertainties in the field of aircraft design are published in [3, 25].

The simulation of probabilistic models is a complex discipline, and the complexity affects the software implementation as well. A design of a general software system to simulate probabilistic models needs to place emphasis on a separation and abstraction of concerns so that a reuse of software units can be exploited as much as possible. Software components [26–30] accurately fit in this context: a software component is understood as a software unit specified by its interface. A component can interchange another component when the functional specifications of the interfaces are the same. A distributed software component is a dynamically and locally/remotely bound software component. A dynamic binding means a binding at runtime and implies that a component interchange happens at runtime. This identifies a high flexibility within a distributed component-based software system. A corresponding software system for the simulation of probabilistic models is proposed in [21, 31, 32]. It provides implementations of numerical schemes to obtain statistics directly or to obtain surrogate models. An essential component in that system is the simulator component which stands for a general abstraction of a simulation code for deterministic models. The system is implemented on the basis of the Component Template Library (CTL) [33], which is a C++ template library to realize distributed component-based software systems.

The aircraft design application in this report is a long-term vision of a new class of low-noise cruise-efficient transport aircrafts with short take-off and landing capabilities (CESTOL) and is subject of the Collaborative Research Center (CRC) — Sonderforschungsbereich (SFB) — 880. The benefits of the approaches, which are applied in the various projects of the SFB 880 and take into account the complex interactions in aircraft design, are assessed on the basis of a jointly adopted aircraft design, the so-called SFB 880 reference configuration. This configuration for around 100 passengers and a range of $2,000$ km features an advanced turboprop propulsion combined with active internally blown high-lift flaps (IBF). Each incremental design change, which is motivated by results continually generated by SFB 880 projects, meets the two following conditions: It follows a progress in technology and affects the aircraft design in a smooth and reliable manner. The latter condition is that the aircraft design be robust. Moreover, the used design data and analysis methods exhibit significant uncertainties which are caused by a lack of knowledge and phenomena on the aircraft itself which are not modeled in-depth, yet. For instance, the assumed maximum aircraft lift coefficient provided by the active high-lift system indicates uncertainties, which are introduced through the used flow simulation method. In addition, the coefficient impacts take-off and landing performance. Also, material constants contain measurement uncertainties. Furthermore, during the life cycle of an aircraft its engine characteristics — namely component efficiencies and pressure ratios — change in an uncertain manner because of their aging process. These circumstances motivate the researcher to combine the classic aircraft design and techniques to model and quantify uncertainties for the novel aircraft concept.

Therefore, the deterministic reference model of the aircraft is extended by a stochastic variation of parameters, resulting in a probabilistic model. The probabilistic model is derived and statistics of that model are approximated by the MC method to discuss the robustness of the reference aircraft design. The MC sampling is performed by the distributed component-based software system which is presented in [21, 31, 32] and installed on a Linux cluster. A deterministic sample is simulated by the Preliminary Aircraft Design and Optimization program (PrADO) [34–36] developed by the Institute of Aircraft Design and Lightweight Structures (IFL) of TU Braunschweig. For this purpose PrADO, which is originally a Windows program, was ported to Linux and expanded to the distributed software component coPrADO. A Python code was implemented for an automatic porting of PrADO. coPrADO is a realization of the mentioned simulator component (it matches the functional specifications of the component interface), and, as a consequence, can be bound at runtime to the distributed component-based software system. The robustness analysis of the CESTOL aircraft and the distributed software component coPrADO were already introduced in our publications [21, 37].

PrADO is outlined in Sec. 2 with regard to the deterministic reference model. The probabilistic model of the aircraft design, its simulation, and the corresponding software design are presented in Sec. 3. Some numerical results about the robustness analysis are discussed in Sec. 4. A conclusion is given in Sec. 5.

2 Multidisciplinary Aircraft Design Program PrADO

The Preliminary Aircraft Design and Optimization program — PrADO — [34–36] of the TU Braunschweig has been used as an integral design methodology in the SFB 880 for technology assessment and overall aircraft design studies. PrADO has a modular structure and its core simulates the iterative overall design process according to Fig. 1 for any desired aircraft concept; the core contains independent design modules (MD i), each of which implements a subtask in the design process. The modules communicate with each other only through a data management system (DMS). This provides for a high flexibility due to which PrADO can be adapted to new design problems. The design core in Fig. 1 is enclosed by two additional loops: The first loop enables an automatic variation or optimization of design variables describing the aircraft configuration and the engine concept; the second loop illustrates the possibility of an uncertainty quantification, which is the main focus in this report. In order to have a wide range of applications, the mentioned design modules largely use physics-based models which are not confined to statistics or specific aircraft configurations. The input includes the specification of the transport mission (e.g. payload, range, and desired cruise conditions), a basic parameter description of the configuration layout and the engine concept, and all significant constraints (e.g. permitted take-off and landing distances and the minimum climb angle according to part 25 of the Federal Aviation Regulations — FAR 25). The output of PrADO is a complete aircraft description with a 3D geometry model, a mass breakdown on component level, aerodynamic and engine characteristics, flight data of the characteristic aircraft missions, an analysis of the direct operating costs (DOC), and there will be noise assessment, too.

The loop of the iterative design process begins with the setting of the aircraft geometry — see part 1 in Fig. 2 — which is derived from predefined non-dimensional geometry parameters (e.g. for lifting surfaces these are aspect ratio, taper ratio, thickness-to-chord (T/C) ratios, sweep angle, reference

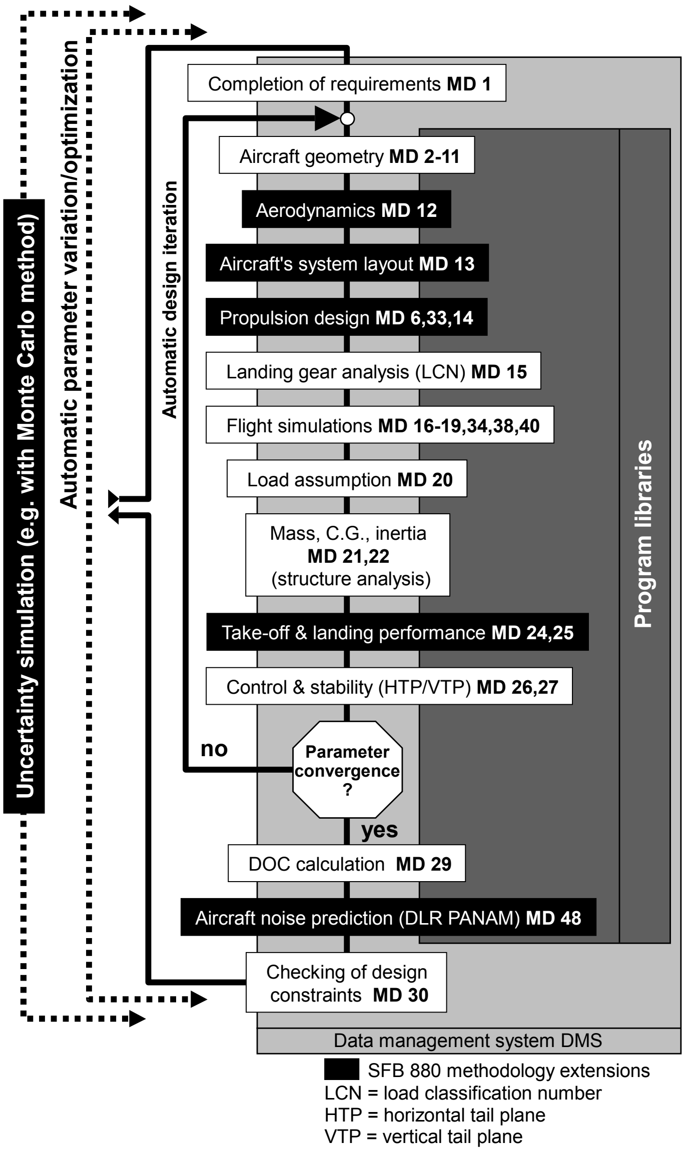


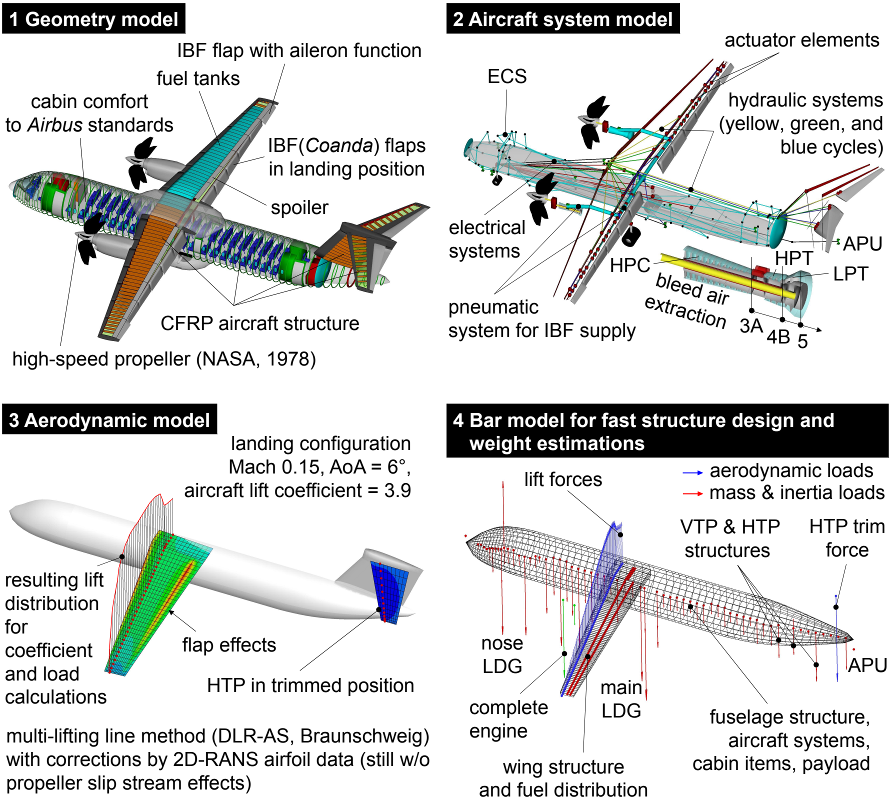
Figure 1: Aircraft design synthesis process of PrADO.

area, and airfoil data) and predetermined body templates for fuselage, fairings, and nacelles. For the fuselage PrADO automatically arranges a cabin layout under consideration of FAR 25 evacuation rules (sizes, positions, and numbers of doors) and determines the body with the smallest wetted surface by a variation of the seat number in the cross-section. The final output is a 3D geometry description which can be recalled by all design modules through interface subroutines. With this technique, for instance, the numerical grids for aerodynamic and structural analyses are derived.

The analysis of the CESTOL aircraft with an active high-lift system requires fundamental methodology extensions. To calculate the aerodynamic characteristics of the short take-off and landing (STOL) aircraft, a multiple lifting-line method with a modeling of deflected IBFs using 2D RANS airfoil data is used, see part 3 in Fig. 2. The desired maximum lift coefficients at take-off and landing are the input of the corresponding design module which then optimizes the flap angle and the necessary mass flow (described by the momentum coefficient) for each IBF plenum. The objective is here to minimize the induced wing drag and the necessary mass flux. At present, propeller slipstream effects are not considered in the calculation of the aerodynamic characteristics. For this, a new approach based on the commercial panel code VSAERO is under development.

The system aspects of the IBFs are included in PrADO through an aircraft system model from Koeppen [38], which was supplemented by a pneumatic line for compressed air supply; the model is shown in part 2 of Fig. 2. The aircraft system design module calculates the masses and the center of gravity positions of the pipes, valves, and distribution elements including the necessary flow properties (flow mass rate, temperature, and total pressure) at the engine bleed ports. These data are directly used for the engine design (with dimensions and engine mass as the outputs) and the calculation of the off-design behavior with and without an operation of the IBF system. Outputs are the thrust and specific fuel consumption maps which are used for the mission analysis and the calculation of the runway lengths. For these tasks, a thermodynamic engine model based on Mattingly [39] is available in the core of PrADO. In this investigation the propeller is not obtained by a calculation method. The necessary data for the overall design investigations — propeller size and mass, gear box mass, and propeller efficiency factor — have been currently taken from excellent NASA investigations [40]. The documented propeller has its highest efficiency factors in the Mach regime between 0.7 and 0.8 (0.861 by $Ma = 0.75$), which fits well to the SFB 880 reference configuration. A 2% reduction of propeller efficiency has been assumed additionally to consider propeller losses through installation effects.

Further calculations in the PrADO process are typical for all kinds of configurations. The aircraft empty weight is calculated by a structural sizing



AoA	=	angle of attack	HTP	=	horizontal tail plane
APU	=	auxiliary power unit	IBF	=	internally blown flap
CFRP	=	carbon-fiber-reinforced plastic	LDG	=	landing gear
ECS	=	environmental control system	LPT	=	low pressure turbine
HPC	=	high pressure compressor	VTP	=	vertical tail plane
HPT	=	high pressure turbine			

Figure 2: PrADO modeling for the assessment of the new CESTOL configuration with high-lift devices based on IBFs.

procedure. A fast approach concerning less computing time is based on a bar model of the primary aircraft structure which is shown for the investigated CESTOL aircraft in part 4 of Fig. 2. Necessary load situations for the structural design are determined by the rules of FAR 25 for maneuver and gust cases. Flight simulations enable to estimate the required fuel mass for the standard missions of a transport aircraft and the take-off and landing performance; here, the important input are the aerodynamic and engine maps. The design modules at the end of an iteration review the size of the empennage and adapt the reference areas of the horizontal and vertical tail plane in the case of missing controllability and/or stability; for this, flight dynamic models are used.

3 A Probabilistic Model and its Simulation

A probabilistic model for an aircraft design is derived from a deterministic reference model to estimate the robustness of the reference model towards quantities of interest. The robustness is proposed to be analysed through the consideration of stochastic properties on these quantities. Quantities of interest are, for instance, the DOC or the maximum runway distances.

The probabilistic model is described in Sec. 3.1, the computation of stochastic properties is outlined in Sec. 3.2. The software design with an emphasis on PrADO, its porting, and its realization as the distributed software component coPrADO is depicted in Sec. 3.3.

3.1 From a Deterministic Model to a Probabilistic One

Parameters of a deterministic reference model of an aircraft are stochastically varied and the effect of these input variations is computed through a simulation of the varied model. A more precise formulation follows in the next passages.

A deterministic aircraft design in the preliminary design phase is described here by a parameterized model with n parameters. To simplify matters, a parameter is assumed to be a real number, so that the parameters can be represented by a vector $\boldsymbol{a} := (a_1, \ldots, a_n) \in A \subset \mathbb{R}^n$. An iterative simulation process of preliminary sizing starts with an initial guess $\boldsymbol{a} = \boldsymbol{a}_0 \in A$ and operates on the parameterized model until the parameters converge. While the initial guess does not need to correspond to a valid aircraft design (valid means technically feasible) the converged parameters correspond to a valid one.

In the deterministic case the preliminary design phase results in a deterministic reference model of the aircraft with parameters $\boldsymbol{a}_{ref} \in A_c$, where set $A_c \subset A$ contains the parameter combinations of all possible valid (converged) aircraft designs. However, the reference model differs from the finally manufactured aircraft. These differences are now taken into account through a stochastic variation of $\boldsymbol{a}_{ref}$ by random vector $\Delta\boldsymbol{a} : \Omega \to \Delta A$ consisting of mutually independent random variables, where Ω is the set of elementary events corresponding to a probability space $\mathcal{P} := (\Omega, \mathcal{F}, P)$ with σ-algebra $\mathcal{F} \subseteq 2^\Omega$ and probability measure P, and $\Delta A \subset \mathbb{R}^n$ is a meaningfully chosen set of variations. The stochastically varied reference model is described by

$\boldsymbol{a}(\omega) := \boldsymbol{a}_{ref} + \Delta\boldsymbol{a}(\omega)$ with $\omega \in \Omega$. A sample of $\boldsymbol{a}(\omega)$, which results from a sample of the stochastic input variation $\Delta\boldsymbol{a}(\omega)$, does not necessarily correspond to a converged design, and as a consequence has to be simulated. The simulation leads to the corresponding converged sample, which is here assigned to be the sample of random vector $\boldsymbol{a}_c(\omega)$. In other words, $\boldsymbol{a}_c(\omega)$ represents the converged aircraft designs according to the stochastically described input variation of the reference model $\boldsymbol{a}_{ref}$ and is the output of simulation process $s : \boldsymbol{a}(\omega) \mapsto \boldsymbol{a}_c(\omega)$. $\boldsymbol{a}_c(\omega)$ may also be written in dependence on its stochastic input variation: $\boldsymbol{a}_c(\Delta\boldsymbol{a})$.

3.2 Determination of Stochastic Quantities of a Probabilistic Model

Stochastic quantities of a probabilistic model like stochastic moments or probability density functions are described by integrals of the form

$$\mathcal{I}_n := \int_\Omega f(\boldsymbol{a}_c(\omega))\, \mathrm{d}P(\omega) = \int_{\boldsymbol{\Delta A}} f(\boldsymbol{a}_c(\Delta\boldsymbol{a}))\, \mathrm{d}P(\Delta\boldsymbol{a}) \tag{1}$$

where f defines an appropriate function. When, for instance, the second stochastic moment is desired f simply squares each component of its argument. The stochastic dimension n of the probabilistic model is usually high.

Widely-used numerical schemes for high-dimensional integration are Monte Carlo (MC) [41–44] and Quasi-Monte Carlo (QMC) methods [41, 44], latin hypercube sampling [12, 45], and the Smolyak algorithm [46, 47]. While MC methods are based on random numbers, QMC methods are based on number-theoretic point sequences. A latin hypercube sampling chooses its samples from an elimination of rows and columns in a regularly partitioned sample space. The Smolyak algorithm provides a high-dimensional integration scheme through a sparse tensor product of one-dimensional quadrature rules. All these numerical schemes approximate the integral in Eq. (1) by the sum

$$\sum_{i=1}^{N} w_i \cdot f(\boldsymbol{a}_c(\Delta\boldsymbol{a}_i)) \;\approx\; \mathcal{I}_n \tag{2}$$

with sample-points $\{\Delta\boldsymbol{a}_i\}_{i\in\{1,\dots,N\}}$ and corresponding weights $\{w_i\}_{i\in\{1,\dots,N\}}$.

3.3 Software Implementation

The following preparatory works were necessary for a simulation of the probabilistic model through the distributed component-based software system in [21, 31, 32] on a Linux cluster. Deterministic samples are simulated by PrADO [34–36], which is originally a Windows program. A Python code was designed to enable an automated porting of PrADO to Linux. After PrADO was available on the Linux cluster, the distributed software component coPrADO [21] was implemented; it interfaces PrADO and provides a remote execution of PrADO. coPrADO is a realization of the simulator component on which the distributed component-based software system operates, and is bound to that system at runtime. Many instances of coPrADO may be created to gain the most from the distributed system. Fig. 3 depicts these preparatory works.

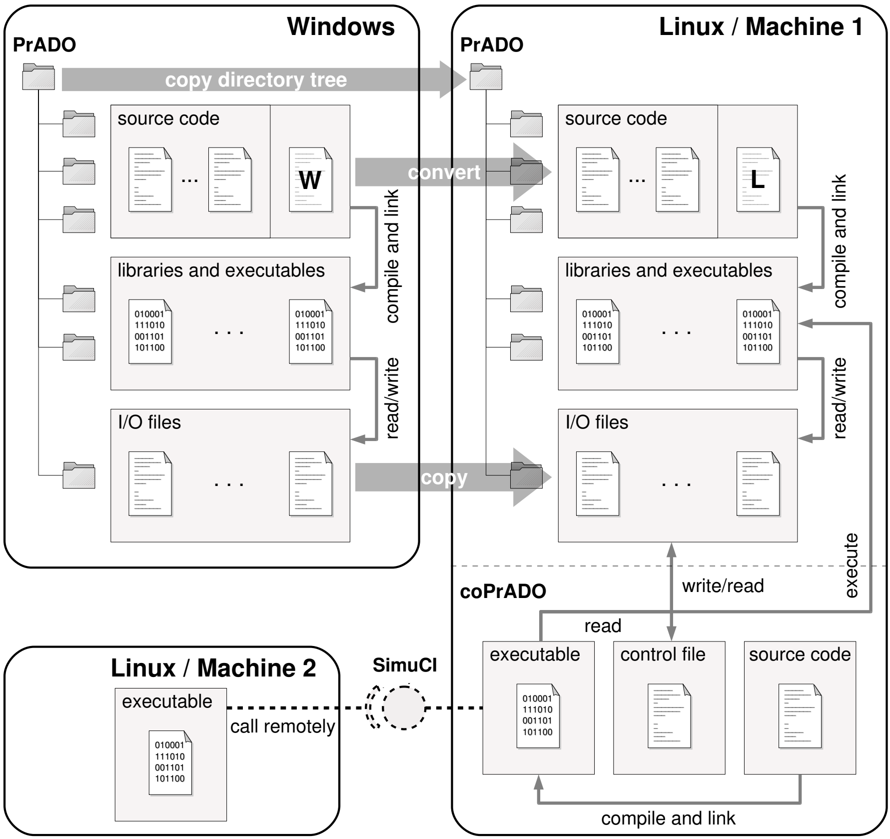


Figure 3: Systematic depiction: PrADO, its porting to Linux, and its expanding to the distributed software component coPrADO.

The porting of PrADO from Windows to Linux is described in Sec. 3.3.1, the distributed software component coPrADO to enable remote calls of PrADO is discussed in Sec. 3.3.2.

3.3.1 Porting of PrADO from Windows to Linux

PrADO is mostly a Fortran code. To simplify matters it is understood as a set of source code files, a set of libraries and executables, which result from the compilation and linkage of the sources, and a set of I/O files. The I/O files are in ASCII format, and store the setting of the parameterized model of an aircraft on which PrADO operates during its iterative simulation process until a converged aircraft model is obtained.

The Python code for the automated porting is lightweight, consisting of about 750 lines of code (commentaries and configuration file inclusive). It is not a distributed application so that PrADO in its Windows version — here abbreviated by W-PrADO — has to be available on the Linux machine, at least during the porting. First of all the Python code copies the directory tree of W-PrADO to the root directory of the desired Linux version of PrADO — abbreviated by L-PrADO —, see Fig. 3. Then the source code files of W-PrADO are converted to the Linux environment. In this process, for instance, the delimiting character of directory paths, the keyword for environmental variables, and package inclusions within the source code are adapted. The conversion is specified in a configuration file which is initially read by the Python code. The configuration file allows extensions of the conversion process without an adaptation of the Python code; that is advantageous when extensions are required for newer versions of PrADO. The source code file `os_dos_dby.f90` of W-PrADO is not converted but replaced; it contains operating system specific functions of higher complexity. This file is indicated by a "**W**" in Fig. 3 for the Windows version and an "**L**" for the Linux version.

When the source code files are converted (or replaced) they are compiled and linked to the libraries and executables of L-PrADO. The I/O files of W-PrADO are simply copied to the L-PrADO directory.

3.3.2 coPrADO — The Distributed Component-Based Realization of PrADO

The distributed component-based software system to simulate the probabilistic model interfaces with the already mentioned simulator component, or, more specifically, it operates on the interface of that component. The simulator component is a generalization of a simulation code for a deterministic model. Different interfaces for a simulator component are available, specified by the requirements of different numerical schemes for the simulation of a probabilistic model [21].

The component interface of interest for this report is called `SimuCI`, see Listing 1; it belongs to numerical schemes which are based on a sampling. It only provides method `set_params` for the setting of the parameters considered to be uncertain, method `solve` for simulating the appropriate deterministic model, and method `get_state` for obtaining the output.

The arguments of type `array<real>` are vectors of real values, and are called by reference. The type `real` is a generic type which, for instance, may be specified by `double` or `float`; as a consequence, the component interface is a generic one which induces further flexibility. This kind of generic component interface is a recent idea [48,49], also discussed in the literature under the term of parametric polymorphism for software components, and is supported by the CTL. Method `solve` returns an integer which tells whether the simulation was successful or not. The deterministic setting of the uncertain parameters is simply passed by a vector of `real`. The identification of the actual parameters — that means the assignment of a real value to the corresponding parameter — may be specified in a so-called control file [21]. The path to that control file is provided to the constructor of `SimuCI` by the argument of type `string`.

```

#define CTL_ClassTmpl SimuCI, ( real ), 1
#include CTL_ClassBegin
    #define CTL_Constructor1 ( const string ), 1
    #define CTL_Method1 void, set_params, ( const array<real> ), 1
    #define CTL_Method2 int4, solve, (), 0
    #define CTL_Method3 void, get_state, ( array<real> ) const, 1
#include CTL_ClassEnd
```

Listing 1: The component interface `SimuCI` in CTL syntax.

The distributed software component coPrADO implements the interface `SimuCI` so that it can be bound at runtime to the distributed component-based software system. coPrADO reads its control file initially to understand for which parameters it receives values through method `set_params`, see Fig. 3. When that method is called coPrADO writes each parameter value to the correct position in the I/O files of PrADO. In that way PrADO's parameterized model of an aircraft is newly set and PrADO's iterative simulation process can be applied to obtain a converged aircraft design. The simulation is performed by method `solve` for which coPrADO executes PrADO. Method `get_state` reads the output of interest from the I/O files and provides it. The identification of the output of interest is also specified here in the control file.

A main routine of the distributed component-based software system is not explicitly shown in Fig. 3 but can be understood as the displayed executable on Linux machine 2.

4 Numerical Experiment

A probabilistic model of the deterministic SFB 880 reference design of a CESTOL aircraft is now simulated; the robustness of the reference design is analysed through MC sampling.

The deterministic reference model and its probabilistic expansion are described in Sec. 4.1, an extraction of the simulation results for the robustness analysis is presented in Sec. 4.2.

4.1 The Deterministic Reference Model and its Probabilistic Model

The requirements for the aircraft focused on in this report are summarized in Table 1 (SL and ISA are abbreviations for sea level and international standard atmosphere). The aircraft is a long-term vision of a new class of low-noise cruise-efficient transport aircrafts with STOL capabilities for the use on airports with short runways; at present there are no such aircrafts in airline operations. If it is possible to realize take-off and landing on 800 m runways nearly 74% of the present European ICAO-certified airports could be used (ICAO stands for International Civil Aviation Organization). This would mean a doubling of the possible destinations compared to the current situation.

First year of operation		2025
Certification according to FAR 25		
Comfort standard of Airbus A320		
Flight with maximum payload (= design case):		
	Range	$2,000$ km
	Payload (100 passengers with baggage and $2,200$ kg additional freight)	$12,000$ kg
Flight with 100 passengers and without additional freight:		
	Range	$\geq 2,800$ km
Flight with maximum fuel:		
	Range is determined by the available tank volume in the wing box	
Cruise conditions (optimized with respect to minimum DOC):		
	Initial cruise altitude	$10,600$ m
	Cruise Mach number	0.74
FAR 25 take-off and landing distances (SL/ISA)		≤ 800 m

Table 1: Design requirements for the SFB 880 aircraft.

Fig. 4 displays the deterministic reference design which had been derived by a classic design approach without considering data and method uncertain-

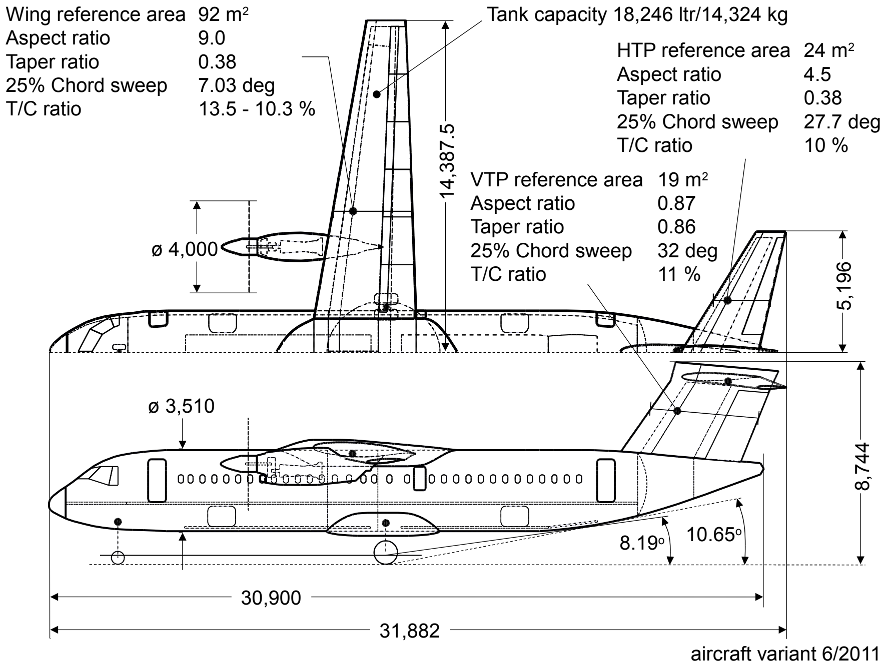


Figure 4: Reference model of the aircraft. All lengths are in mm.

ties. The aircraft configuration has a high wing arrangement with turboprop engines mounted in front of the wing. The efficient propulsion system in combination with a carbon primary structure leads to a reduced take-off and landing weight, which helps to achieve the STOL requirements. The pivotal element is, however, the internally blown high-lift system that enables large flow turning and hence 50% larger lift coefficients compared to conventional multi-element solutions. The extraction of compressed air takes place behind the high pressure compressor (HPC), the high pressure turbine (HPT), and the low pressure turbine (LPT) of the engine to enable a safe IBF operation in all low-speed flight phases. The wing with IBF system is designed for maximum wing lift coefficients of 3.5 during take-off and 4.5 during the final phases of the landing. That determines the necessary mass flow rates of 13.1 kg/s during the take-off with moderate flap angles of 45°, and of 14.4 kg/s during the landing with flap angles of around 70°. The engine is designed to deliver the maximum IBF flow rates as necessary in a situation with one engine inoperative (OEI). The wing has a simple tapered planform with a low leading edge sweep of 10° to reduce wing weight and manufacturing costs.

In addition, the reduced sweep improves the usable maximum lift coefficient, which is approximately 11% higher compared to a wing with 28° sweep (common for current aircrafts of the 100-seat class). The empennage is arranged as a classic T-tail to move the horizontal stabilizer out of the propeller steam. The main design data are listed in Table 2 (USD/skm stands for US Dollar per revenue seat-kilometer).

Description	value	unit of measurement
Wing reference area	92	m^2
Maximum engine thrust (SL/ISA)		
IBF system off	$2 \times 93,240$	N
IBF system on/landing configuration	$2 \times 89,166$	N
Maximum engine shaft power (SL/ISA)		
IBF system off	$2 \times 8,870$	kW
IBF system on/landing configuration	$2 \times 8,396$	kW
Bleed air for		
the aircraft systems	0.88	kg/s
the IBF flaps (max., landing configuration)	14.444	kg/s
Total	15.324	kg/s
Specific fuel consumption (SFC) during cruising	$4.593 \cdot 10^{-2}$	g/N/h
Maximum lift coefficient in landing configuration	4.5	
Lift coefficient during cruising	0.4604	
Lift-to-drag (L/D) ratio during cruising	14.491	
Operating empty weight (OEW):		
Mass of the wing	3,499	kg
Mass of the fuselage	5,408	kg
Mass of the propulsion system	4,542	kg
Mass of the IBF system	205	kg
Total	23,928	kg
Total fuel mass (design case)	4,722	kg
Maximum take-off weight (MTOW)	40,650	kg
Maximum landing mass	38,855	kg
Range with maximum fuel	8,013	km
FAR 25 take-off distance (SL,ISA)	779	m
FAR 25 landing distance (SL,ISA)	759	m
Approach speed in landing configuration	50.36	m/s
DOC (design case)	$7.951 \cdot 10^{-2}$	USD/skm

Table 2: Reference model: Key features of the reference model (SFB 880 aircraft variant from June 2011).

The cruise condition and the wing planform have also been optimized. The objective is here to minimize the DOC. The optimal cruise condition is found at Mach 0.74 and an altitude of 10.6 km. The high Mach number improves the DOC due to an increase of revenue seat-kilometers and is limited by the propeller characteristics. The cruise altitude is limited by the constraint that the cruise lift coefficient should not exceed a value greater than 0.5. This constraint had been introduced as a design margin to cause, in combination with the low sweep, no problems with the transonic drag rise. Fig. 5 shows the wing loading as a function of the thrust loading for the CESTOL design problem, which depicts the reasons for the proper wing size. The figure also already includes sampling data of the uncertainty quantification; however, the corresponding discussion is postponed to the next section. It is interesting that the required landing distance of 800 m can be achieved with a relatively high wing loading of 489.5 kg/m^2 (wing reference area of 83.5 m^2), which makes the STOL aircraft competitive to conventional aircrafts relating to cruising aerodynamics. This is a clear advantage of the IBF approach. On the other hand, the necessary aircraft thrust is driven by the take-off requirement of 800 m runway length. For this reason, and because of a wing loading of 489.5 kg/m^2, the aircraft needs approximately 17% more maximum take-off thrust, which also shows the negative effect of bleed air extraction for the IBF system. Therefore, the overall optimum lies by a slightly larger wing reference area of 92 m^2, which corresponds to a wing loading of 441.8 kg/m^2. The optimization of other wing planform parameters (aspect ratio, taper ratio, and T/C ratios, see Fig. 4) is dominated by their effects on the wing structural loads; the goal is a light wing and therefore a cheaper aircraft.

The parameters which mainly influence the relevant design characteristics of the STOL aircraft are varied through a stochastic description. The variation of each parameter is described by a uniformly distributed random variable. The parameters and their variation ranges are specified in Table 3. They represent uncertainties within the methods used in the overall process to calculate the aircraft properties (e.g. using a simple aerodynamic vortex model with 2D RANS airfoil data to calculate the maximum wing lift with circulation control), and also uncertainties in the input like material data for structure sizing or engine component data (e.g. efficiencies of compressors and turbines) for the calculation of the engine performance maps. Both kinds of uncertainties are understood here to be epistemic ones; for a discussion of epistemic and aleatory uncertainties please see for instance [50].

4.2 Robustness Analysis Through a Monte Carlo Sampling

The basic MC method was applied to compute statistics and distribution functions for technical parameters of the probabilistic model, for instance engine shaft power, aircraft and component weights, cruise L/D ratio, and landing field length. Scatter plots of the parameters were generated as well to identify their sensitivities on the design of the STOL aircraft. Some results and the derived robust design are consecutively presented.

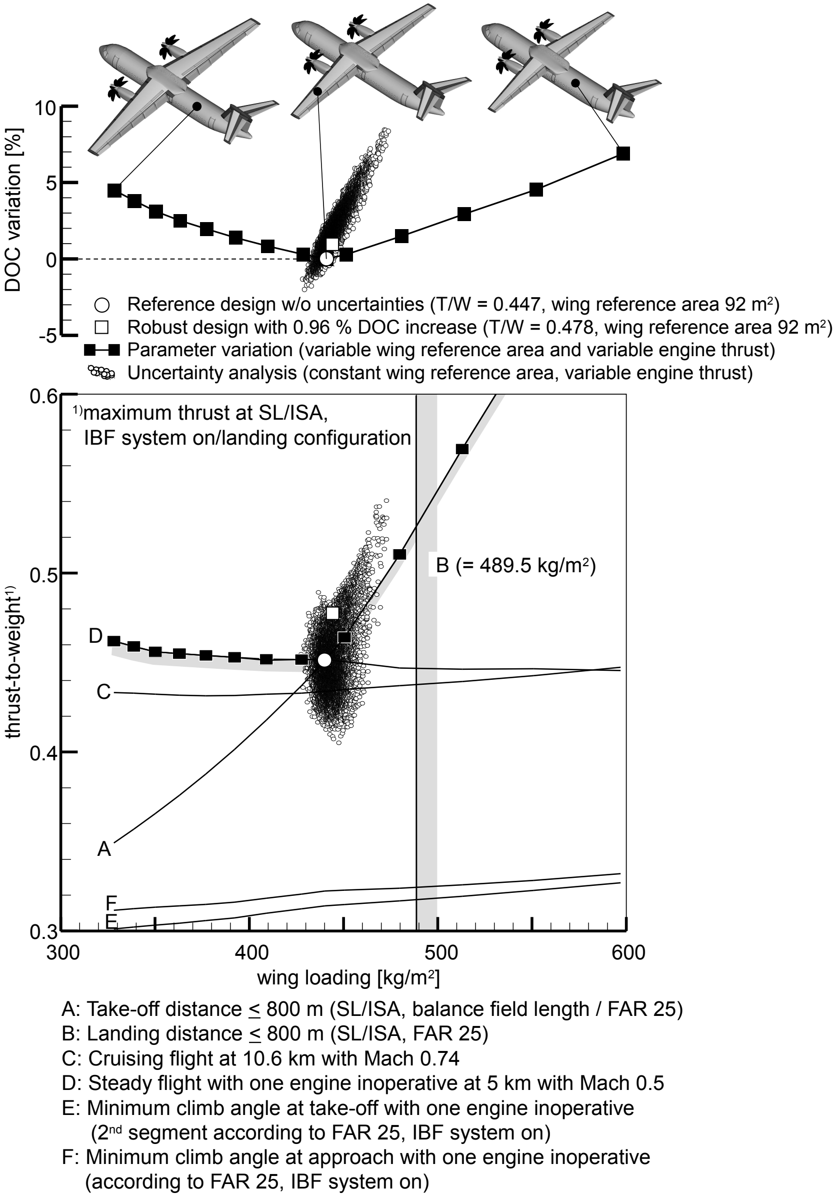


Figure 5: Wing loading versus thrust-to-weight (T/W) ratio of the SFB 880 CESTOL aircraft variant from June 2011.

No.	parameter	[min,max] in [%]
Structure:		
1	Mass of the wing	$[-5, 5]$
2	Mass of the fuselage	$[-5, 5]$
Aerodynamics:		
3	Lift coefficient	$[-5, 5]$
4	Lift-induced drag coefficient	$[-5, 5]$
5	Zero-lift drag coefficient (mainly viscous drag)	$[-5, 5]$
Propulsion system:		
6	Mass of the propulsion system	$[-5, 5]$
7	Polytropic efficiency of the HPC	$[-3, 3]$
8	Polytropic efficiency of the HPT	$[-3, 3]$
9	Polytropic efficiency of the LPT	$[-3, 3]$
10	Combustion efficiency	$[-3, 0]$
11	Total pressure ratio of the HPC	$[-3, 3]$
12	Maximum turbine inlet temperature (TET)	$[-3, 3]$
Blown flap system:		
13	Mass of the blown flap system	$[-50, 100]$
14	Mass flow rate for the blown flap system	$[-5, 5]$

Table 3: Varied parameters with minimum and maximum variations in percentage.

For the uncertainty quantification $12,187$ MC samples were simulated by twenty-four 3.0 GHz processors on a Linux cluster. The mean of the runtime for one simulation — which is equal to a full run of PrADO — was 4.2 h with a standard deviation of 1.1 h. Table 4 presents the mean and the standard deviation of some parameters. The central limit theorem [51] was applied to estimate the relative error of the considered stochastic moments. With a high probability the relative error of the means is below 10^{-3}, and the one of the standard deviations is below 10^{-1}.

Fig. 6a displays the probability density function and its cumulative distribution function for the maximum engine take-off thrust, which also represents the necessary thrust of the aircraft to conform to the design requirements (see Table 1). The maximum engine thrust is at the most 7.59% higher than the reference case for 90% of the parameter realizations. From the viewpoint of the overall design only a slight increase of the necessary engine power is required to obtain a robust aircraft design. That is all the more surprising as a coupling between power plant and aerodynamics exists due to the powered lift system. The same applies for the necessary landing field length, see Fig. 6b. The demanded landing field length is not met for only 0.04% of the samples, with a maximum deviation of only around five meters.

Parameter	reference model	means	std. deviation	unit of measurement
Maximum engine shaft power (IBF system on/landing configuration)	8,396	8,538	357	kW
OEW	23,928	24,067	363	kg
Total fuel mass (design case)	4,722	4,837	232	kg
MTOW	40,650	40,904	540	kg
L/D ratio during cruising	14.491	14.519	$3.613 \cdot 10^{-1}$	–
FAR 25 landing field length	759	761	10	m
DOC (design case)	$7.951 \cdot 10^{-2}$	$8.016 \cdot 10^{-2}$	$1.077 \cdot 10^{-3}$	USD/skm

Table 4: Means and standard deviations for some parameters of the probabilistic model in comparison to the reference model.

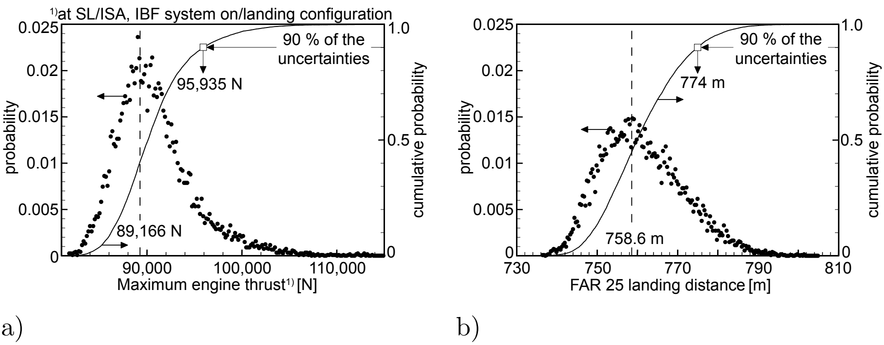
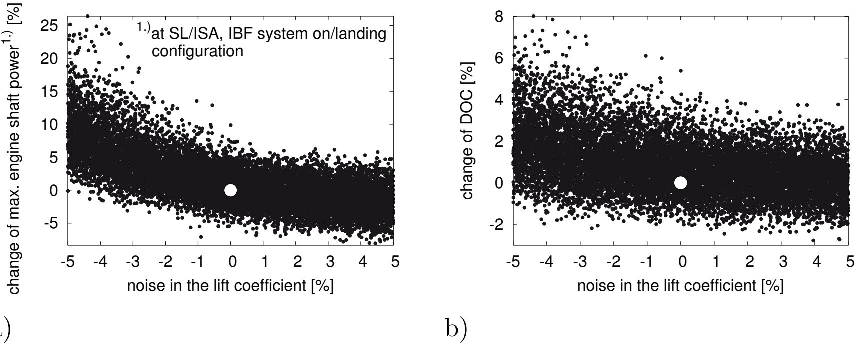


Figure 6: Probability density functions and cumulative distribution functions of important design parameters of the probabilistic model (dashed lines represent the parameters of the reference model).

The scatter plot of the (maximum) lift coefficient and the maximum engine shaft power is presented in Fig. 7a. It is obvious that a reduction of the lift leads to a stronger increase of the demanded power. That is indirectly a consequence of a convention which was introduced into the PrADO process. When the requirement of the allowed take-off distance (see Table 1) is not satisfied the thrust is appropriately adapted so that the requirement will be satisfied. As a consequence, the DOC is also influenced by that convention, see Fig. 7b: A higher usable lift coefficient requires a less powerful engine which also means a less expensive engine; the displayed reduction of the DOC arises from the effect of depreciation allowance and interest. A redesign of the CESTOL aircraft in consideration of the described results of the uncertainty quantification leads to the desired robust aircraft design. In Fig. 5 it can be observed that the robust design — depicted by the white square — exhibits a certain distance to the critical design boundaries — lines A and D; it is located more in the allowed design space, whereby design reserves towards unexpected events arise. Through this the aircraft configuration obtains its desired robustness. Corresponding changes of main aircraft parameters in relation to the classic design approach are presented in Fig. 8. The robust design has a higher maximum take-off weight due to the larger engine. The increase of the weight also determines a slightly higher block fuel requirement of around 1.69%. The utilization of the higher engine power which became necessary as a result of the uncertainty consideration reduces the take-off length by 3.97%; as a consequence, the larger engine has a positive side effect. The higher aircraft empty weight — due to the heavier propulsion group — implicates a higher landing weight. That is the reason for the slightly higher landing distance of 0.52% of the robust design, which, however, does not violate the requirement of 800 m. The DOC takes a relatively large increase for the robust design of 0.96%. Over an operating period of fourteen years this percentage leads to a cost increase of about 3.4 million USD.



a) b)

Figure 7: Scatter plots of parameters of the probabilistic model measured in their percentage change; the reference model is marked by a big white circle, respectively.

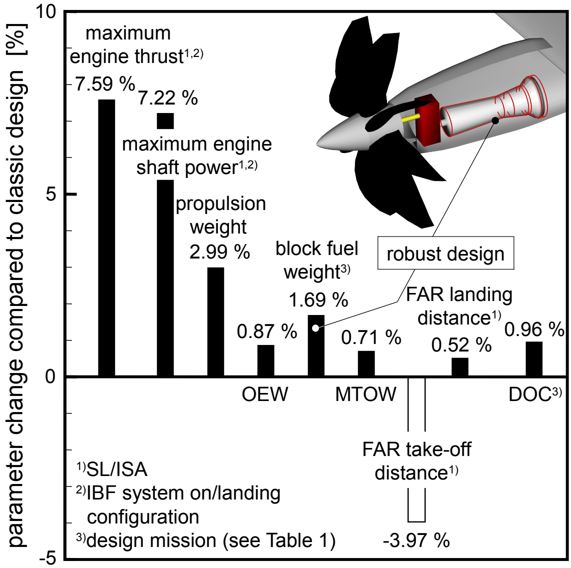


Figure 8: Change of important aircraft parameters through the demand on design robustness (SFB 880 CESTOL aircraft variant from June 2011).

5 Conclusion

The robustness of a deterministic reference model of a CESTOL aircraft with an active high-lift device was analysed here. For this purpose a probabilistic model was derived and stochastic quantities like means, standard deviations, and distribution functions were computed through an MC sampling, and scatter plots were generated. In consideration of the computational results the reference model was rated to be robust. From the viewpoint of the overall aircraft design the surprising result of the uncertainty quantification was that in comparison with the reference CESTOL design only 7.59% of thrust increase (7.22% more shaft power) is necessary to cover 90% of the error cases.

A sample of the probabilistic model was simulated by the PrADO code for a preliminary sizing. The MC sampling was performed through a distributed component-based software system installed on a Linux cluster. For this PrADO was ported from Windows to Linux because it is originally a Windows program. A Python code was implemented for an automatic porting. The Linux version of PrADO was then realized as a distributed software component with the name coPrADO.

The MC sampling is expensive because many samples are required for a higher accuracy. As a consequence, a surrogate model is planned to be approximated on the basis of less samples, which is to provide accurate results in a fast post-processing step. Both sparse and low-rank approaches should be tried in this context. The computed surrogate model should be used as a cost-efficient proxy for simulating different distributions for the input variations of the aircraft design.

Acknowledgements

Financial support has been provided by the German Research Foundation (Deutsche Forschungsgemeinschaft — DFG) in the framework of the Collaborative Research Center (Sonderforschungsbereich — SFB) 880 "Fundamentals of high-lift for future commercial aircraft".

References

[1] Jaeger, L., Gogu, C., Segonds, S., and Bes, C., "Aircraft multidisciplinary design optimization under both model and design variables uncertainty," *Journal of Aircraft*, Vol. 50, No. 2, 2013, pp. 528–538. doi: 10.2514/1.C031914

[2] Wynn, D. C., Grebici, K., and Clarkson, P. J., "Modelling the evolution of uncertainty levels during design," *International Journal on Interactive Design and Manufacturing*, Vol. 5, No. 3, 2011, pp. 187–202. doi: 10.1007/s12008-011-0131-y

[3] Yao, W., Chen, X., Luo, W., van Tooren, M., and Guo, J., "Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles," *Progress in Aerospace Sciences*, Vol. 47, No. 6, 2011, pp. 450–479. doi: 10.1016/j.paerosci.2011.05.001

[4] Mavris, D. N., DeLaurentis D. A., Bandte, O., and Hale, M. A., "A stochastic approach to multi-disciplinary aircraft analysis and design," *36th Aerospace Sciences Meeting & Exhibit*, Georgia Institute of Technology, Reno, 1998, pp. 1–15.

[5] Siva, C., Senthil Murugan, M., and Ganguli, R., "Uncertainty quantification in helicopter performance using Monte Carlo simulations," *Journal of Aircraft*, Vol. 48, No. 5, 2011, pp. 1503–1511. doi: 10.2514/1.C000288

[6] Neufeld, D., Chung, J., and Behdinian, K., "Aircraft conceptual design optimization considering fidelity uncertainties," *Journal of Aircraft*, Vol. 48, No. 5, 2011, pp. 1602–1612. doi: 10.2514/1.C031312

[7] Danowsky, B. P., Chrstos, J. R., and Klyde, D. H., "Evaluation of aeroelastic uncertainty analysis methods," *Journal of Aircraft*, Vol. 47, No. 4, 2010, pp. 1266–1273. doi: 10.2514/1.47118

[8] Mavris, D. N., DeLaurentis D. A., and Soban, D. S., "Probabilistic assessment of handling qualities constraints in aircraft preliminary design," *36th Aerospace Sciences Meeting & Exhibit*, Georgia Institute of Technology, Reno, 1998, pp. 1–13.

[9] DeLaurentis D. A., Mavris, D. N., and Schrage, D. P., "System synthesis in preliminary aircraft design using statistical methods," *20th Congress of the International Council of the Aeronautical Sciences*, Georgia Institute of Technology, Sorrento, 1996, pp. 1–13.

[10] Mavris, D. N., and DeLaurentis, D. A., "A probabilistic approach for examining aircraft concept feasibility and viability," *Aircraft Design*, Vol. 3, No. 2, 2000, pp. 79–101. doi: 10.1016/S1369-8869(00)00008-2

[11] Mavris, D. N., and Macsotai, N. I., and Roth, B. A., "A probabilistic design methodology for commercial aircraft engine cycle selection," *3rd World Aviation Congress and Exposition*, Georgia Institute of Technology, Anaheim, 1998.

[12] Díaz, J., and Hernández, S., "Uncertainty quantification and robust design of aircraft components under thermal loads," *Aerospace Science and Technology*, Vol. 14, No. 8, 2010, pp. 527–534. doi: 10.1016/j.ast.2010.04.004

[13] Kirby M. R., and Mavris, D. N., "Forecasting technology uncertainty in preliminary aircraft design," *4th World Aviation Congress and Exposition*, San Francisco, 1999, pp. 1–12.

[14] Xiu, D., and Karniadakis, G. E., "The Wiener-Askey polynomial chaos for stochastic differential equations," *SIAM Journal on Scientific Computing*, Vol. 24, No. 2, 2002, pp. 619–644. doi: 10.1137/S1064827501387826

[15] Ghanem, R. G., and Spanos, P. D., *Stochastic finite elements: a spectral approach.* Dover Publications, Mineola, 2003.

[16] Najm, H. N., "Uncertainty quantification in fluid flow," *Turbulent Combustion Modeling*, Fluid Mechanics and Its Applications, Vol. 95, Springer, Berlin, 2011, pp. 381–407.

[17] Blatman, G., and Sudret, B., "Adaptive sparse polynomial chaos expansion based on least angle regression," *Journal of Computational Physics*, Vol. 230, No. 6, 2011, pp. 2345–2367. doi: 10.1016/j.jcp.2010.12.021

[18] Nobile, F., Tempone, R., and Webster, C. G., "A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data," *SIAM Journal on Numerical Analysis*, Vol. 46, No. 5, 2008, pp. 2309–2345. doi: 10.1137/060663660

[19] Doostan, A., and Owhadi, H., "A non-adapted sparse approximation of PDEs with stochastic inputs," *Journal of Computational Physics*, Vol. 230, No. 8, 2011, pp. 3015–3034. doi: 10.1016/j.jcp.2011.01.002

[20] Bieri, M., and Schwab, C., "Sparse high order FEM for elliptic SPDEs," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 13–14, 2009, pp. 1149–1170. doi: 10.1016/j.cma.2008.08.019

[21] Krosche, M., "A generic component-based software architecture for the simulation of probabilistic models," Ph.D. Dissertation, Institute of Scientific Computing, Carl-Friedrich-Gauß-Fakultät, TU Braunschweig, Braunschweig, 2013.

[22] Nouy, A., "Recent developments in spectral stochastic methods for the numerical solution of stochastic partial differential equations," *Archives of Computational Methods in Engineering*, Vol. 16, No. 3, 2009, pp. 251–285. doi: 10.1007/s11831-009-9034-5

[23] Matthies, H. G., and Zander, E., "Solving stochastic systems with low-rank tensor compression," *Linear Algebra and its Applications*, Vol. 436, No. 10, 2012, pp. 3819–3838. doi: 10.1016/j.laa.2011.04.017

[24] Litvinenko, A., and Matthies, H. G., "Uncertainties quantification and data compression in numerical aerodynamics," *Proceedings in Applied Mathematics and Mechanics*, Vol. 11, No. 1, 2011, pp. 877–878. doi: 10.1002/pamm.201110425

[25] Zang T. A., Hemsch, M. J., Hilburger, M. W., Kenny, S. P., Luckring, J. M., Maghami P., Padula S. L., and Stroud, W. J., *Needs and opportunities for uncertainty-based multidisciplinary design methods for aerospace vehicles*, NASA/TM-2002-211462, 2002.

[26] Czarnecki, K., and Eisenecker, U. W., *Generative programming: methods, tools, and applications*, Addison-Wesley, Boston, 2000.

[27] Szyperski, C., *Component software: beyond object-oriented programming*, Addison-Wesley, Boston, 2002.

[28] Kaur, A., and Singh Mann, K., "Component based software engineering," *International Journal of Computer Applications*, Vol. 2, No. 1, 2010, pp. 105–108. doi: 10.5120/605-855

[29] Koziolek, H., "Performance evaluation of component-based software systems: a survey," *Performance Evaluation*, Vol. 67, No. 8, 2010, pp. 634–658. doi: 10.1016/j.peva.2009.07.007

[30] Lau, K.-K., and Wang, Z., "Software component models," *IEEE Transactions on Software Engineering*, Vol. 33, No. 10, 2007, pp. 709–724. doi: 10.1109/TSE.2007.70726

[31] Krosche, M., and Hautefeuille, M., "Simulation and solution of stochastic systems with a component-based software design," *Proceedings in Applied Mathematics and Mechanics*, Vol. 7, No. 1, 2007, pp. 2140001–2140002. doi: 10.1002/pamm.200700067

[32] Krosche, M., and Matthies, H. G., "Component-based software realisations of Monte Carlo and stochastic Galerkin methods," *Proceedings in Applied Mathematics and Mechanics*, Vol. 8, No. 1, 2008, pp. 10765–10766. doi: 10.1002/pamm.200810765

[33] Niekamp, R., "CTL manual for Linux/Unix for the usage with C++," Institute of Scientific Computing, TU Braunschweig, Braunschweig, 2012.

[34] Heinze, W., Österheld, C. M., and Horst, P., "Multidisziplinäres Flugzeugentwurfsverfahren PrADO — Programmentwurf und Anwendung im Rahmen von Flugzeug-Konzeptstudien," *Jahrbuch der DGLR-Jahrestagung 2001*, Hamburg, 2001, pp. 12.

[35] Werner-Westphal, C., Heinze, W., and Horst, P., "Multidisciplinary integrated preliminary design applied to unconventional aircraft configurations," *Journal of Aircraft*, Vol. 45, No. 2, 2008, pp. 581–590. doi: 10.2514/1.32138

[36] Werner-Spatz, C., Heinze, W., Horst, P., and Radespiel, R., "Multidisciplinary conceptual design for aircraft with circulation control high-lift systems," *CEAS Aeronautical Journal*, Vol. 3, No. 2, 2012, pp. 145–164. doi: 10.1007/s13272-012-0049-5

[37] Krosche, M., and Heinze, W., "An aircraft design with uncertain parameters: a robustness analysis through a component-based software

system," *SFB 880 — Fundamentals of High-Lift for Future Commercial Aircraft, Biennial Report*, CFF Forschungsberichte 2013–03, TU Braunschweig — Campus Forschungsflughafen, Braunschweig, 2013, pp. 191–203.

[38] Koeppen, C., "Method for model-based estimations of system masses in aircraft pre-design," *66th Annual International Conference on Mass Properties Engineering*, SAWE, 2007.

[39] Mattingly, J. D., Heiser, W. H., and Daley, D. H., *Aircraft engine design*, AIAA, New York, 1987, Appendix H Turboprop Engine, pp. 507–531.

[40] Baum, J. A., Dumais, P. J., Mayo, M. G., Metzger, F. B., Shenkman, A. M., and Walker, G. G., "Prop-fan data support study," *Hamilton Standard, Division of United Technologies Corporation and NASA Ames Research Center*, NASA CR–152141, Moffett Field, 1978.

[41] Caflisch, R. E., "Monte Carlo and Quasi-Monte Carlo methods," *Acta Numerica*, Vol. 7, Cambridge, 1998, pp. 1–49.

[42] Liu, J. S., *Monte Carlo strategies in scientific computing*, Springer, New York, 2001.

[43] Madras, N., *Lectures on Monte Carlo methods*, American Mathematical Society, Providence, 2002.

[44] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical recipes in C++: the art of scientific computing*, Cambridge University Press, Cambridge, 2002.

[45] McKay, M. D., Beckman, R. J., and Conover, W. J., "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

[46] Gerstner, T., and Griebel, M., "Dimension-adaptive tensor-product quadrature," *Computing*, Vol. 71, No. 1, 2003, pp. 65–87. doi: 10.1007/s00607-003-0015-5

[47] Petras, K., "On the Smolyak cubature error for analytic functions," *Advances in Computational Mathematics*, Vol. 12, No. 1, 2000, pp. 71–93. doi: 10.1023/A:1018904816230

[48] Oancea, C. E., and Watt, S. M., "An architecture for generic extensions," *Science of Computer Programming*, Vol. 76, No. 4, 2011, pp. 258–277. doi: 10.1016/j.scico.2009.09.008

[49] Chicha, Y., Lloyd, M., Oancea, C. E., and Watt, S. M., "Parametric polymorphism for computer algebra software components," *6th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, Timisoara, 2004, pp. 119–130.

[50] Der Kiureghian, A., Ditlevsen, O., "Aleatory or epistemic? Does it matter?," *Structural Safety*, Vol. 31, No. 2, 2009, pp. 105–112. doi: 10.1016/j.strusafe.2008.06.020

[51] Adams, M., and Guillemin, V., *Measure theory and probability*, Birkhäuser, Boston, 1996.

2011-06	R. van Glabbeek, U. Goltz, J.-W. Schicke	On Causal Semantics of Petri Nets
2011-07	H. Cichos, S. Oster, M. Lochau, A. Schürr	Extended Version of Model-based Coverage-Driven Test Suite Generation for Software Product Lines
2011-08	W.-B. Pöttner, J. Morgenroth, S. Schildt, L. Wolf	An Empirical Performance Comparison of DTN Bundle Protocol Implementations
2011-09	H. G. Matthies, A. Litvinenko, O. Pajonk, B. V. Rosić and E. Zander	Parametric and Uncertainty Computations with Tensor Product Representations
2011-10	B. V. Rosić, A. Kučerová, J. Sýkora, A. Litvinenko, O. Pajonk and H. G. Matthies	Parameter Identification in a Probabilistic Setting
2011-11	M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies and E. Zander	Efficient Analysis of High Dimensional Data in Tensor Formats
2011-12	S. Oster	A Semantic Preserving Feature Model to CSP Transformation
2012-01	O. Pajonk, B. V. Rosić and H. G. Matthies	Deterministic Linear Bayesian Updating of State and Model Parameters for a Chaotic Model

2012-02	B. V. Rosić and H. G. Matthies	Stochastic Plasticity — A Variational Inequality Formulation and Functional Approximation Approach I: The Linear Case
2012-03	J. Rang	An analysis of the Prothero-Robinson example for constructing new DIRK and ROW methods
2012-04	S. Kolatzki, M. Hagner, U. Goltz and A. Rausch	A Formal Definition for the Description of Distributed Concurrent Components — Extended Version
2012-05	M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies and P. Wähnert	Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats
2012-06	S. Mennike	A Petri Net Semantics for the Join-Calculus
2012-07	S. Lity, R. Lachmann, M. Lochau, I. Schaefer	Delta-oriented Software Product Line Test Models — The Body Comfort System Case Study
2013-01	M. Lochau, S. Mennicke, J. Schroeter und T. Winkelmann	Extended Version of 'Automated Verification of Feature Model Configuration Processes based on Workflow Petri Nets'
2013-02	S. Lity, M. Lochau, U. Goltz	A Formal Operational Semantics of Sequential Function Tables for Model-based SPL Conformance Testing
2013-03	L. Giraldi, A. Litvinenko, D. Liu, H. G. Matthies, A. Nouy	To be or not to be intrusive? The solution of parametric and stochastic equations — the "plain vanilla" Galerkin case
2013-04	A. Litvinenko, H. G. Matthies	Inverse problems and uncertainty quantification

2013-05	J. Rang	Improved traditional Rosenbrock-Wanner methods for stiff ODEs and DAEs
2013-06	J. Koslowski	Deterministic single-state 2PDAs are Turing-complete
2014-01	B. Rosić, J. Diekmann	Stochastic Description of Aircraft Simulation Models and Numerical Approaches
2014-02	M. Krosche, W. Heinze	A Robustness Analysis of a Preliminary Design of a CESTOL Aircraft